

## Math Background:

So before going into the diffie-hellman algorithm let's give some prerequisite math background

### • Fundamentals of Power:

- Product law  $\rightarrow x^a \cdot x^b = x^{ab}$
- So we can use the product law in the opposite direction

$2 | a \Rightarrow$  This means a is divisible by 2

$$x^a = x^{\frac{a}{2}} \cdot x^{\frac{a}{2}}$$

$2 \nmid a \Rightarrow$  This means a is not divisible by 2, i.e. 'a' is odd

$$\cancel{x^a}$$

$$a = 1 + (a-1) \quad 2 \mid a-1$$

$$x^a = x^1 \cdot x^{a-1}$$

$$= x^{\frac{(a-1)}{2}} \cdot x^{\frac{(a-1)}{2}}$$

$\Rightarrow$  we can simplify this by writing  $x^a = x^1 \cdot x^{\frac{a-1}{2}} \cdot x^{\frac{a-1}{2}}$

## Fundamentals of Modular Arithmetic:

- First you know

~~Ex. 1.1.2.2 [Basic]~~

$$a \% n = 0 \dots n-1$$

- Mod operator can be distributed smoothly among expression)

$$\text{Addition: } (a+b) \% n = (a \% n + b \% n) \% n$$

$$\text{Multiplication: } (ab) \% n = (a \% n \cdot b \% n) \% n$$

Proof:  $(a+b) \% n = (a \% n + b \% n) \% n$

writing 'a' in terms of n

$$a = qn + r \quad q: \text{quotient}$$

r: remainder

$$\text{E.g. } a = 10 \quad n = 3$$

$$10 = 3 \times 3 + 1 \quad q = 3$$

$$r = 1$$

let's continue:

$$a = q_1 n + r_1 \quad b = q_2 n + r_2$$

→ let's continue proving ~~Euclid's theorem~~

$$(a+b) \% n = (a \% n + b \% n) \% n$$

$$a = q_1 n + r_1 \quad b = q_2 n + r_2$$

$$(a+b) \% n = (q_1 n + r_1 + q_2 n + r_2) \% n$$

$$= [(q_1 + q_2)n + (r_1 + r_2)] \% n$$

Now we know that Mod removes all the cycles from a number, In other words

$$kn \% n = 0 \text{ for } \forall k \in \mathbb{Z}$$

$$\text{So } (q_1 + q_2)n \% n = 0$$

So we can say

$$[(q_1 + q_2)n + (r_1 + r_2)] \% n = \\ (r_1 + r_2) \% n$$

recall that Mod computes the remainder,

so

$$r_1 = a \% n \quad r_2 = b \% n$$

and that concludes our proof that

$$(a \% n + b \% n) \% n = (a+b) \% n$$

→ You can use the same principles to proof that for the case of Multiplication

$$(ab) \% n = (a \% n \cdot b \% n) \% n$$

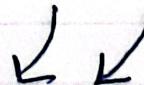
- So we have talked a lot about Modular Arithmetic, now let's combine Mod with Powers

$$a^b \% n \quad \text{we know that}$$

we know that  $a^b = a^{b_2} \cdot c^{b_2}$  iff  $2 \mid b$ , or  $a^b = a \cdot a^{1 \frac{b_1}{2}} \cdot a^{1 \frac{b_2}{2}}$  iff  $2 \nmid b$

$$\text{So } a^b \% n = (a^{b_2} \cdot a^{b_2}) \% n \text{ iff } 2 \mid b$$

$$= (a^{b_2 \% n} \cdot a^{b_2 \% n}) \% n$$



That is a very very Imp Concept to understand about Mod with Powers

~~Fact 2: If we know the base  
 $c^{b_2 \% n}$~~

~~Result~~

$\Rightarrow$  Some Facts about Mod and Power.

We have just shown that

$$a^b \% n = (a^{b/2} \% n \cdot a^{b/2} \% n) \% n$$

iff  $2 \mid b$

AND

$$a^b \% n = (a \% n \cdot a^{1\frac{b}{2}} \% n \cdot a^{1\frac{b}{2}} \% n) \% n$$

iff  $2 \nmid b$

Fact 1: We only need to compute  $a^{b/2} \% n$  to be able to know the value of  ~~$a^b \% n$~~

Fact 2: We can divide  $a^{b/2} \% n$  further to  $b/4 \quad b/4$   
 $a \cdot a$

Given the above 2 facts this implies that we can calculate

$a^b \% n$  in only  $\log_2(b)$  steps

$\Rightarrow$  So  $a^b \% n$  can be computed very efficiently even for very large values for the exponent

## Bonus:

- Here is a pseudo code to show how  $a^b \% n$  can be computed
- The following code is recursive ↗

```
pow-mod(a, b, n) {  
    if (b == 1) // base case  
        return a % n;
```

```
    SubResult = pow-mod(a, b/2, n);  
    SubResult = SubResult * SubResult;
```

```
    if (b % 2 != 0) // odd exponent  
        SubResult = SubResult *  
            (a % n)
```

```
    return SubResult % n;
```

```
}
```